COMP9321:
Data services engineering

Week 9: Classification + project introduction

Semester 2, 2018

By Mortada Al-Banna, CSE UNSW

Machine Learning for Data Analytics

1. **Define** and **Initialize** a Model
2. **Train** your Model (using your training dataset)
3. **Validate** the Model (by prediction using your test dataset)
4. Use it: **Explore** or **Deploy** as a web service
5. **Update** and **Revalidate**

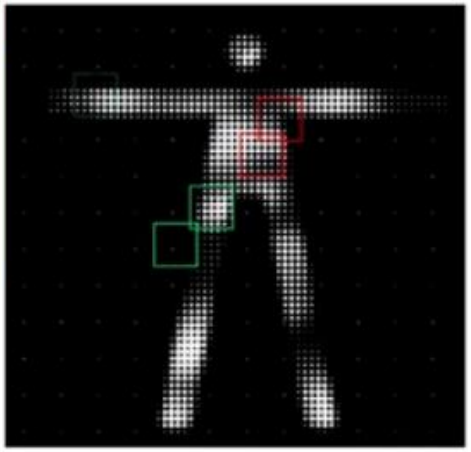
https://www.slideshare.net/SetuChokshi/machine-learning-101-74017810

Classification

- Supervised Learning
- You need the data labelled with the correct answer to train the algorithm
- Trained classifiers then can map input data to a category.

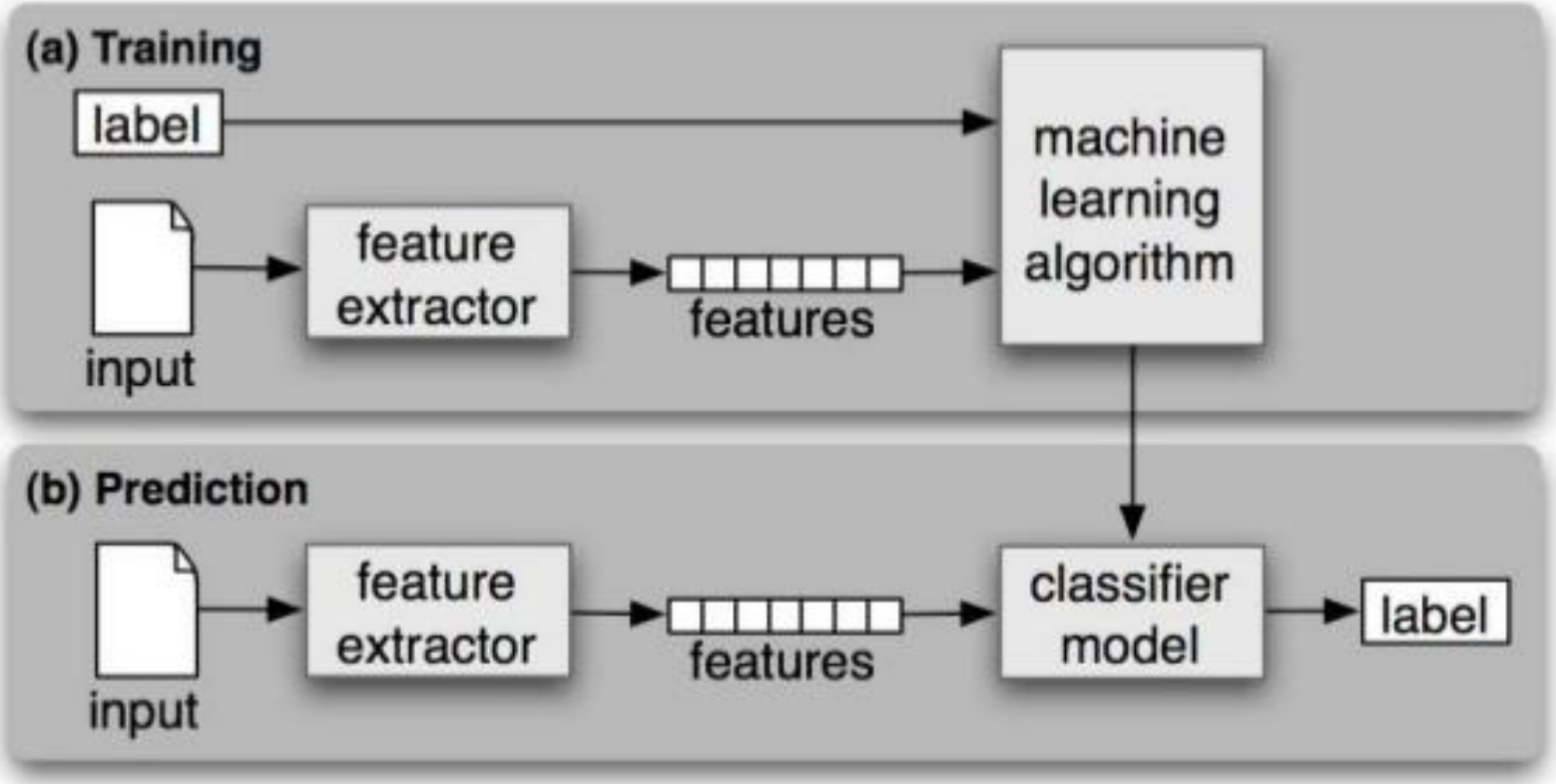
https://www.slideshare.net/brianjlange/machine-learning-in-5-minutes-classification

Classification Examples



Credit: Foundations of Data Science: Prediction and Machine Learning course offered by Berkley
https://courses.edx.org/courses/course-v1:BerkeleyX+Data8.3x+2T2018/course/

Example of a General Flow



https://www.slideshare.net/rahuldausa/introduction-to-machine-learning-38791937

k-Nearest Neighbour (k-NN)

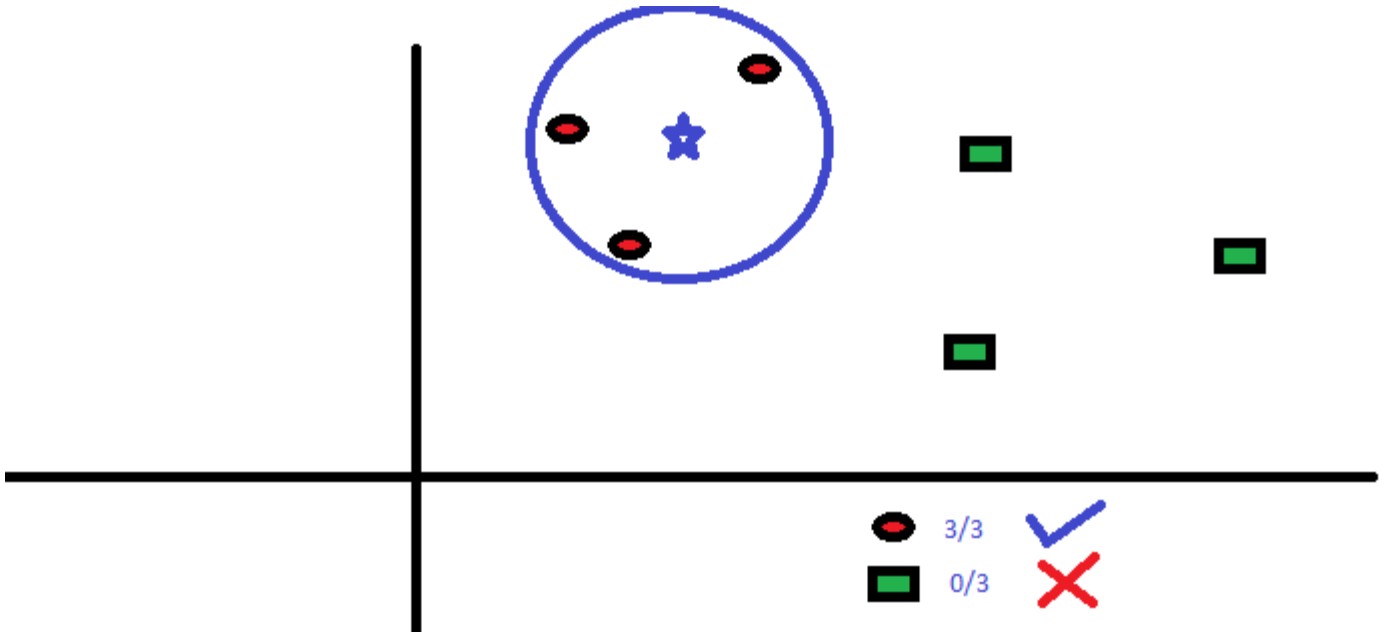
The KNN classifier is a **non parametric** and **instance-based** learning algorithm.

Non-parametric means it makes no explicit assumptions about the functional form of how the prediction is made, avoiding the dangers of mismodeling the underlying distribution of the data.

Instance-based learning means that our algorithm doesn't explicitly learn a model. Instead, it chooses to memorize the training instances which are subsequently used as "knowledge" for the prediction phase. Concretely, this means that only when a query to our database is made (i.e. when we ask it to predict a label given an input), will the algorithm use the training instances to spit out an answer.

https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

k-Nearest Neighbour (k-NN)



https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

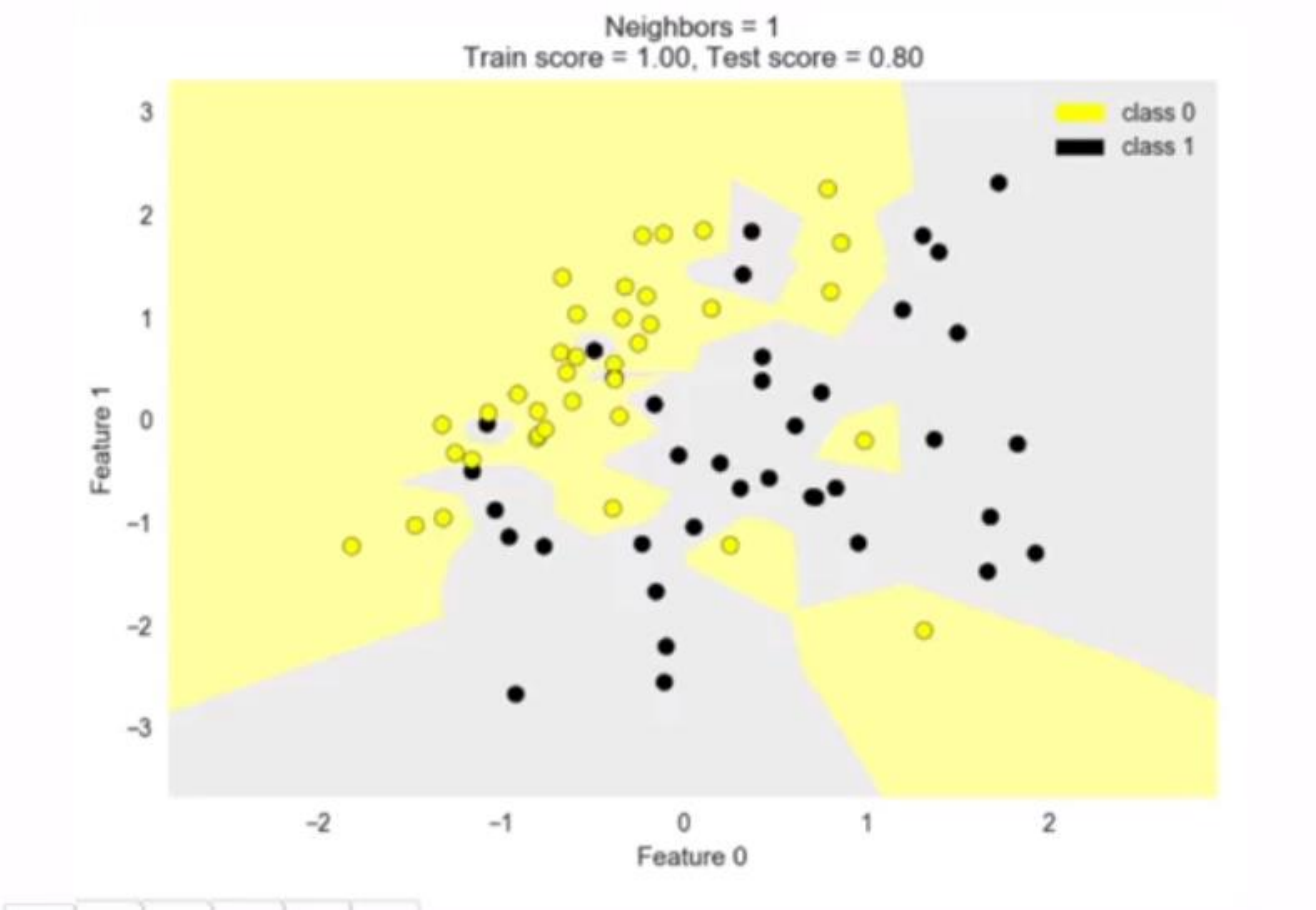
k- Nearest Neighbour Classifier Algorithm

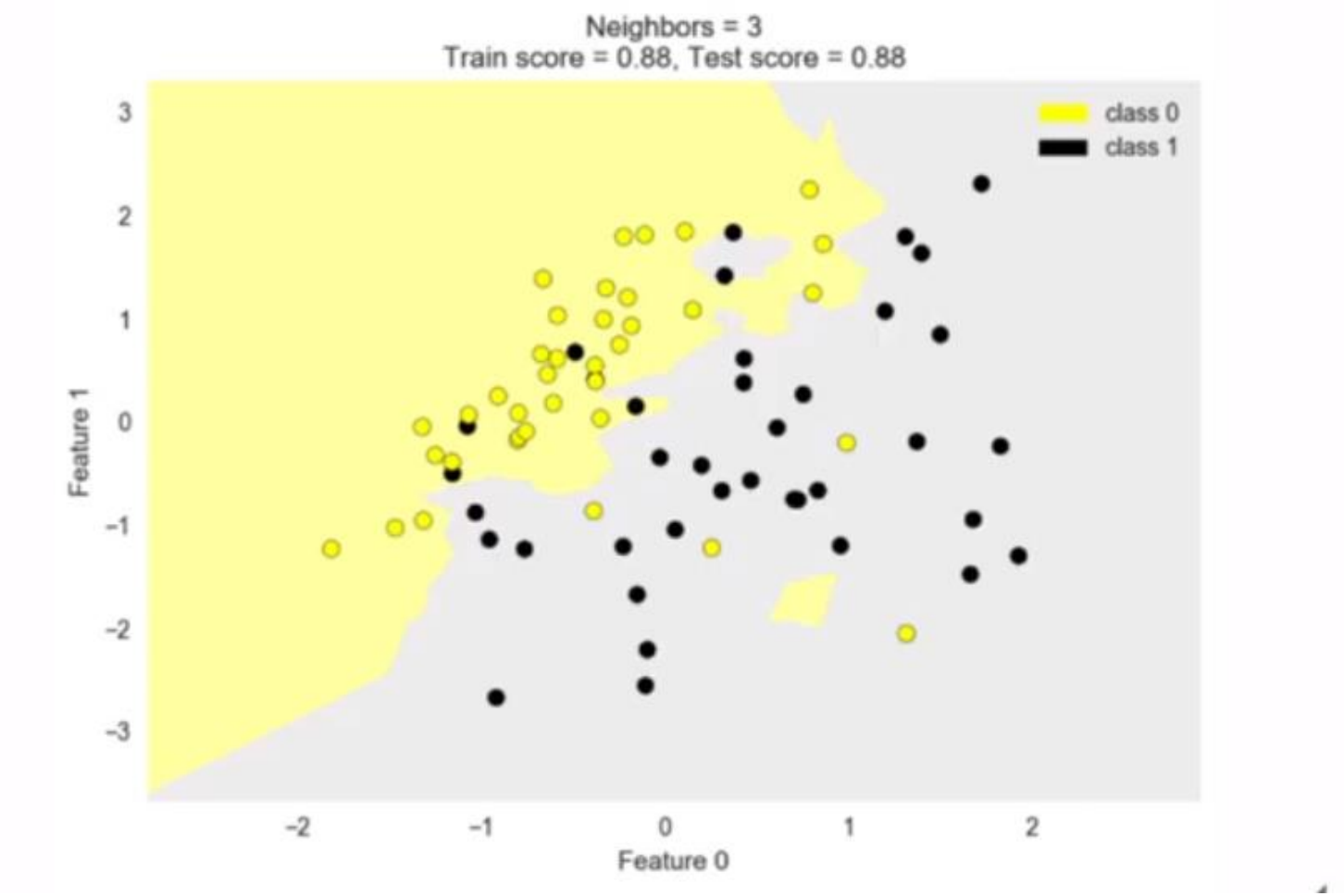
Give a training set X_train with lables y_train and given a new instance x_test to be classified:

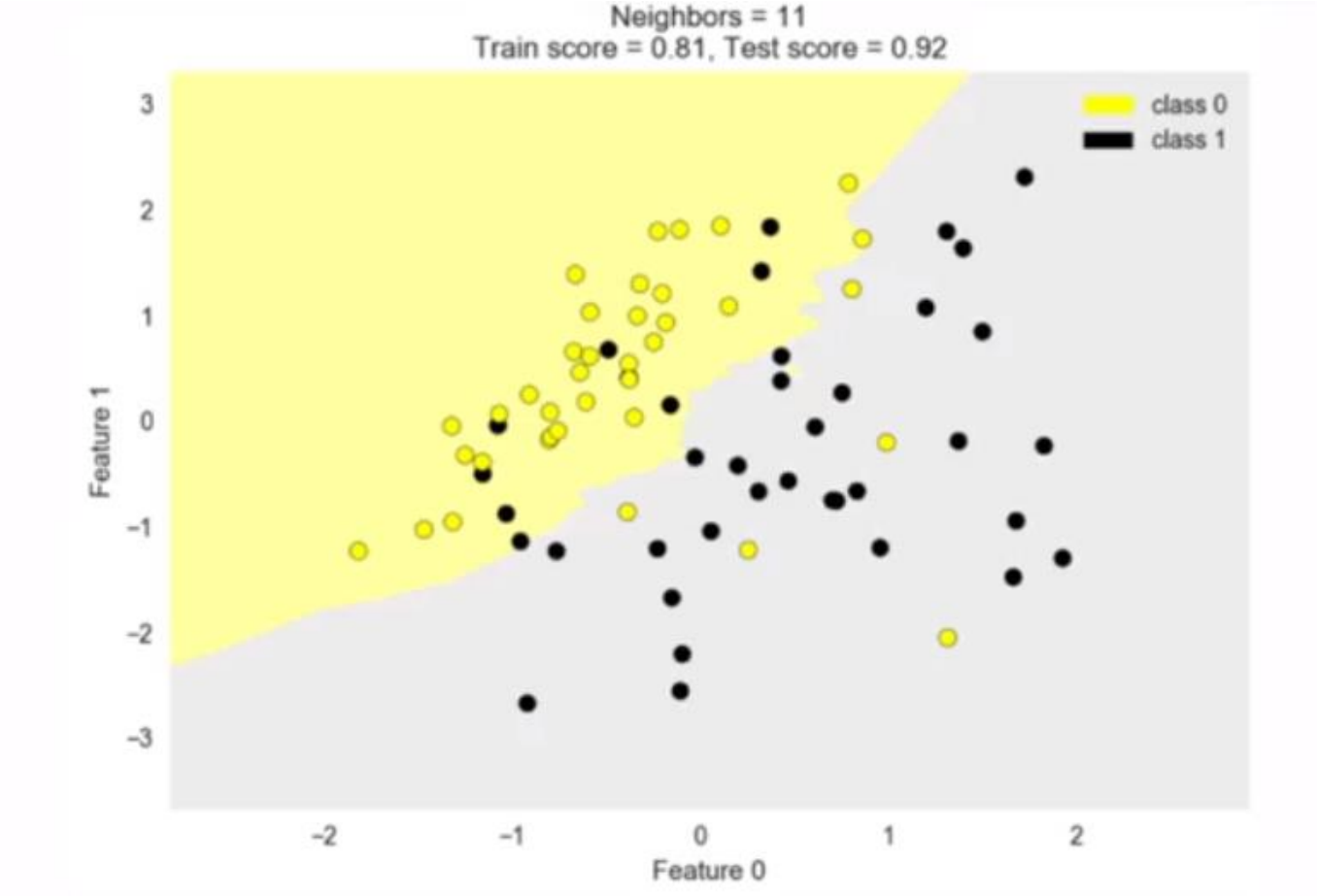
1. Find the most similar instances (let's call then X_NN) to x_test that are in X_train.
2. Get the labels y_NN for the instances in X_NN.
3. Predict the label for x_test by combining the labels y_NN (e.g., using majority rule)

Nearest Neighbour Need Four things Specified

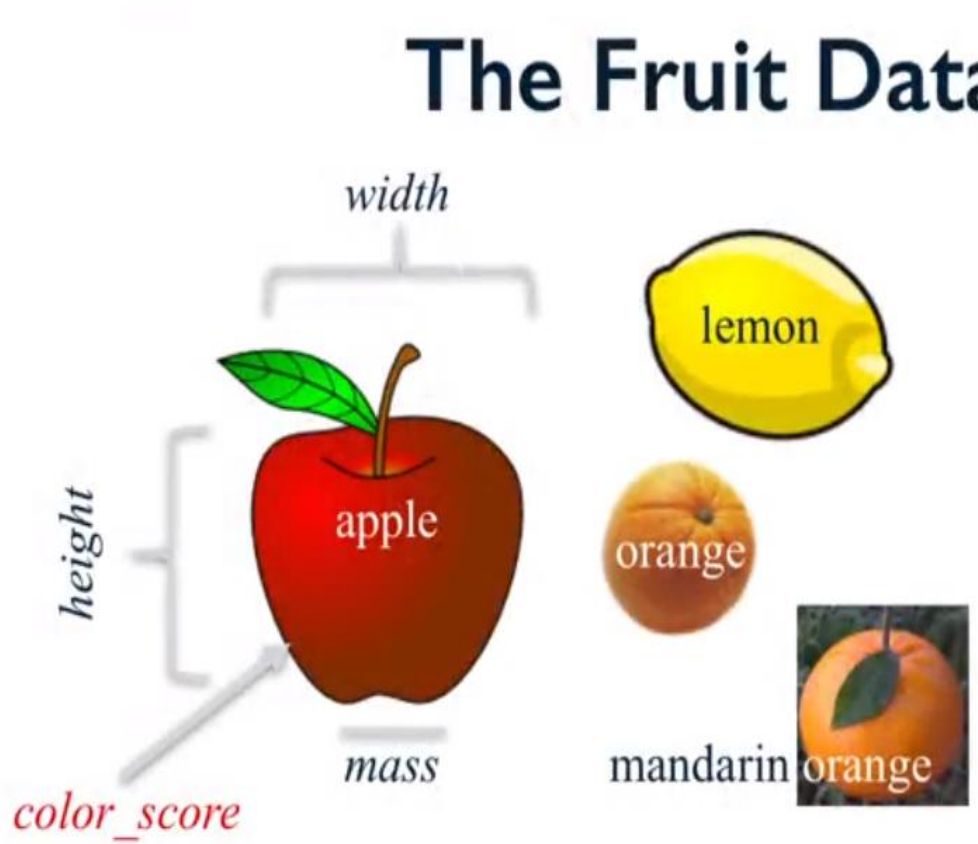
1. A distance Metric (Typically Euclidean)
2. How many nearest neighbours to look at (e.g., Five)
3. Optional Weighting function on the neighbours points (e.g., closer points are weighted higher than farther points)
4. How to aggregate the classes of neighbours points (e.g., simple majority voting)


Neighbors = 1
Train score = 1.00, Test score = 0.80
class 0
class 1
Feature 1
Feature 0
3
2
1
0
−1
−2
−3
−2
−1
0
1
2


Neighbors = 3
Train score = 0.88, Test score = 0.88
class 0
class 1
Feature 1
Feature 0
3
2
1
0
−1
−2
−3
−2
−1
0
1
2


Neighbors = 11
Train score = 0.81, Test score = 0.92
class 0
class 1
Feature 1
Feature 0
3
2
1
0
−1
−2
−3
−2
−1
0
1
2

Classification Data Set Example



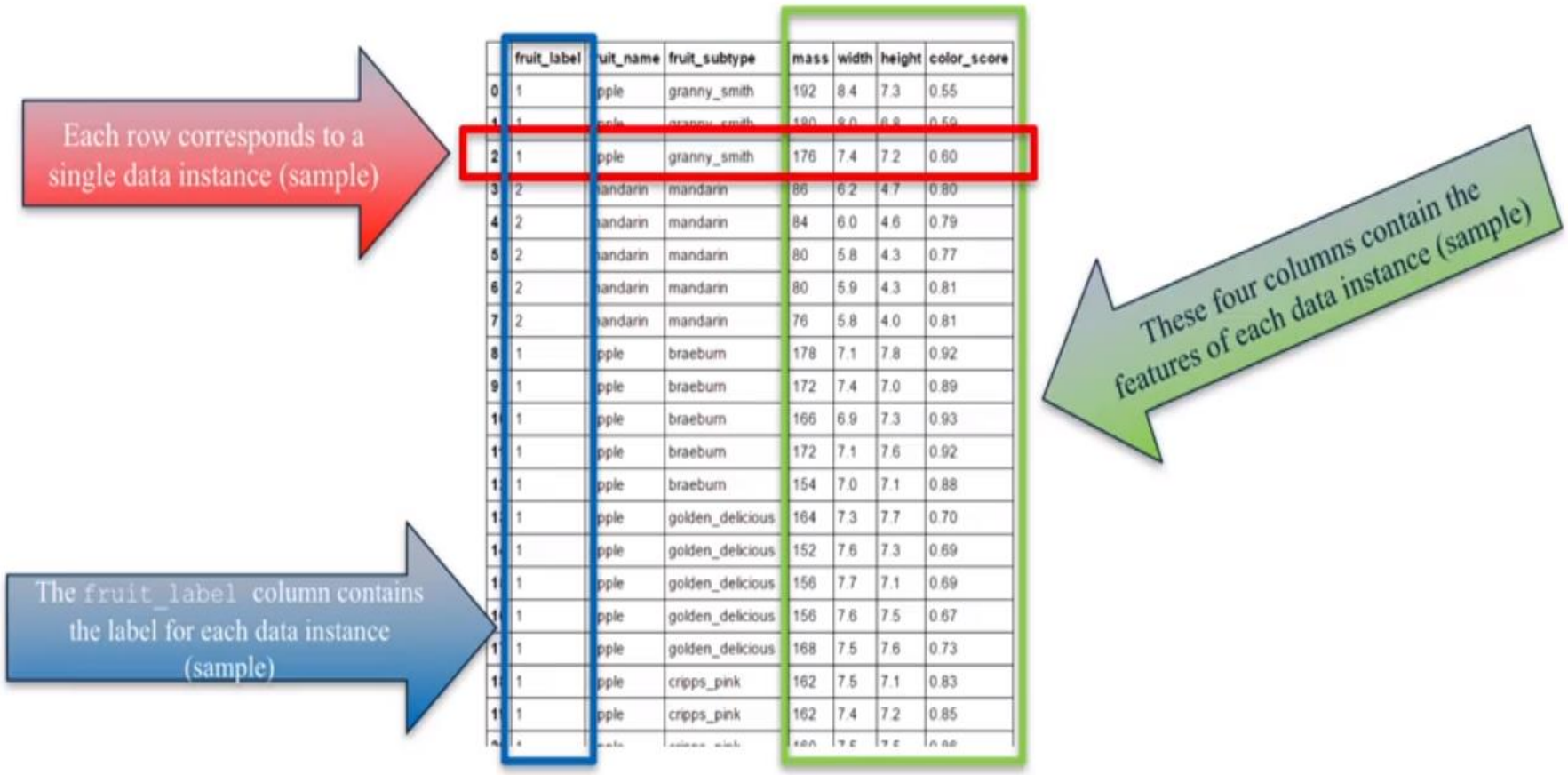
	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59
2	1	apple	granny_smith	176	7.4	7.2	0.60
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79
5	2	mandarin	mandarin	80	5.8	4.3	0.77
6	2	mandarin	mandarin	80	5.9	4.3	0.81
7	2	mandarin	mandarin	76	5.8	4.0	0.81
8	1	apple	braeburn	178	7.1	7.8	0.92
9	1	apple	braeburn	172	7.4	7.0	0.89
10	1	apple	braeburn	166	6.9	7.3	0.93
11	1	apple	braeburn	172	7.1	7.6	0.92
12	1	apple	braeburn	154	7.0	7.1	0.88
13	1	apple	golden_delicious	164	7.3	7.7	0.70
14	1	apple	golden_delicious	152	7.6	7.3	0.69
15	1	apple	golden_delicious	156	7.7	7.1	0.69
16	1	apple	golden_delicious	156	7.6	7.5	0.67

`fruit_data_with_colors.txt`

Credit: Original version of the fruit dataset created by Dr. Iain Murray, Univ. of Edinburgh

Example taken from Applied Machine Learning in Python course by the University of Michigan
https://www.coursera.org/learn/python-machine-learning

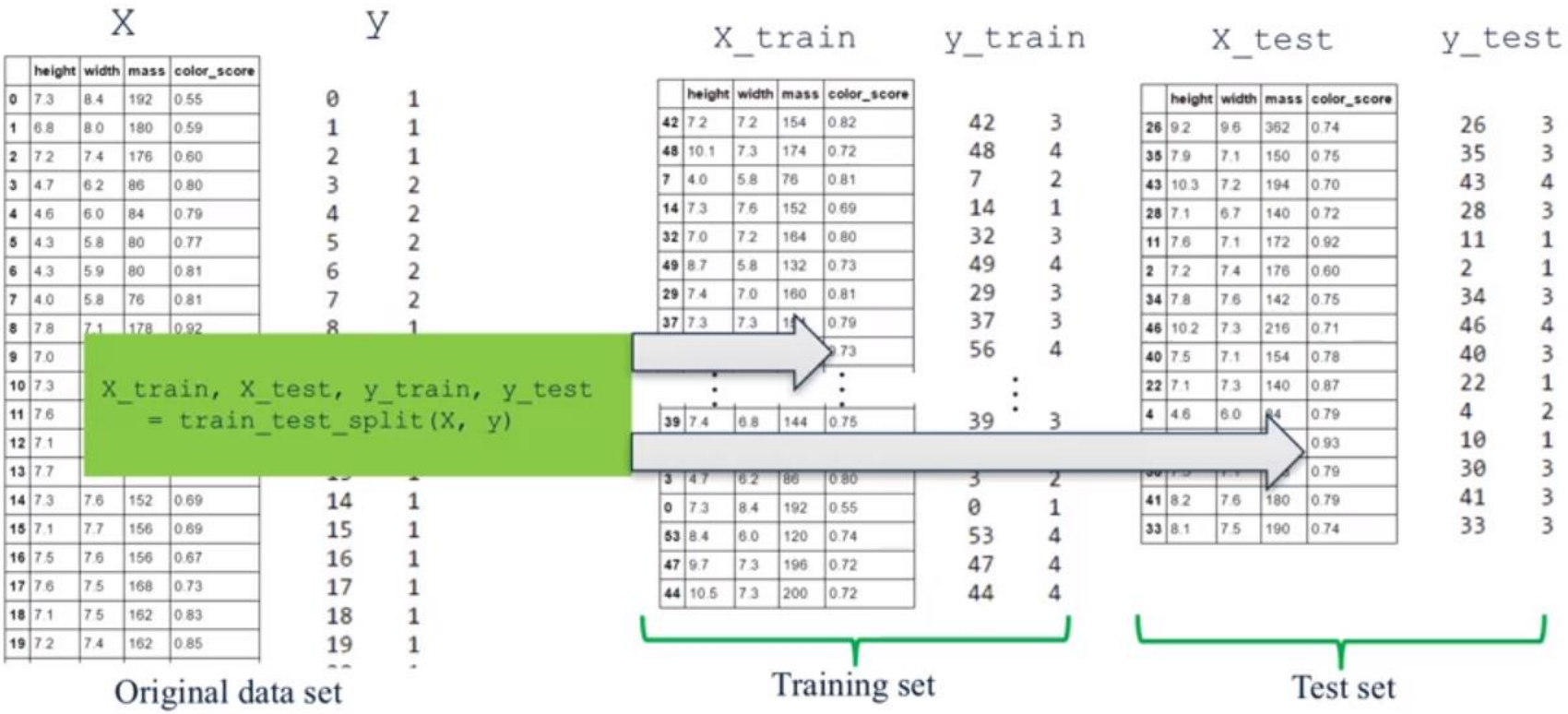
Data As a Table



	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
2	1	apple	granny_smith	176	7.4	7.2	0.60
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79
5	2	mandarin	mandarin	80	5.8	4.3	0.77
6	2	mandarin	mandarin	80	5.9	4.3	0.81
7	2	mandarin	mandarin	76	5.8	4.0	0.81
8	1	apple	braeburn	178	7.1	7.8	0.92
9	1	apple	braeburn	172	7.4	7.0	0.89
10	1	apple	braeburn	166	6.9	7.3	0.93
11	1	apple	braeburn	172	7.1	7.6	0.92
12	1	apple	braeburn	154	7.0	7.1	0.88
13	1	apple	golden_delicious	164	7.3	7.7	0.70
14	1	apple	golden_delicious	152	7.6	7.3	0.69
15	1	apple	golden_delicious	156	7.7	7.1	0.69
16	1	apple	golden_delicious	156	7.6	7.5	0.67
17	1	apple	golden_delicious	168	7.5	7.6	0.73
18	1	apple	cripps_pink	162	7.5	7.1	0.83
19	1	apple	cripps_pink	162	7.4	7.2	0.85

Example taken from Applied Machine Learning in Python course by the University of Michigan
https://www.coursera.org/learn/python-machine-learning

Training Set ad Test Set



Example taken from Applied Machine Learning in Python course by the University of Michigan
https://www.coursera.org/learn/python-machine-learning

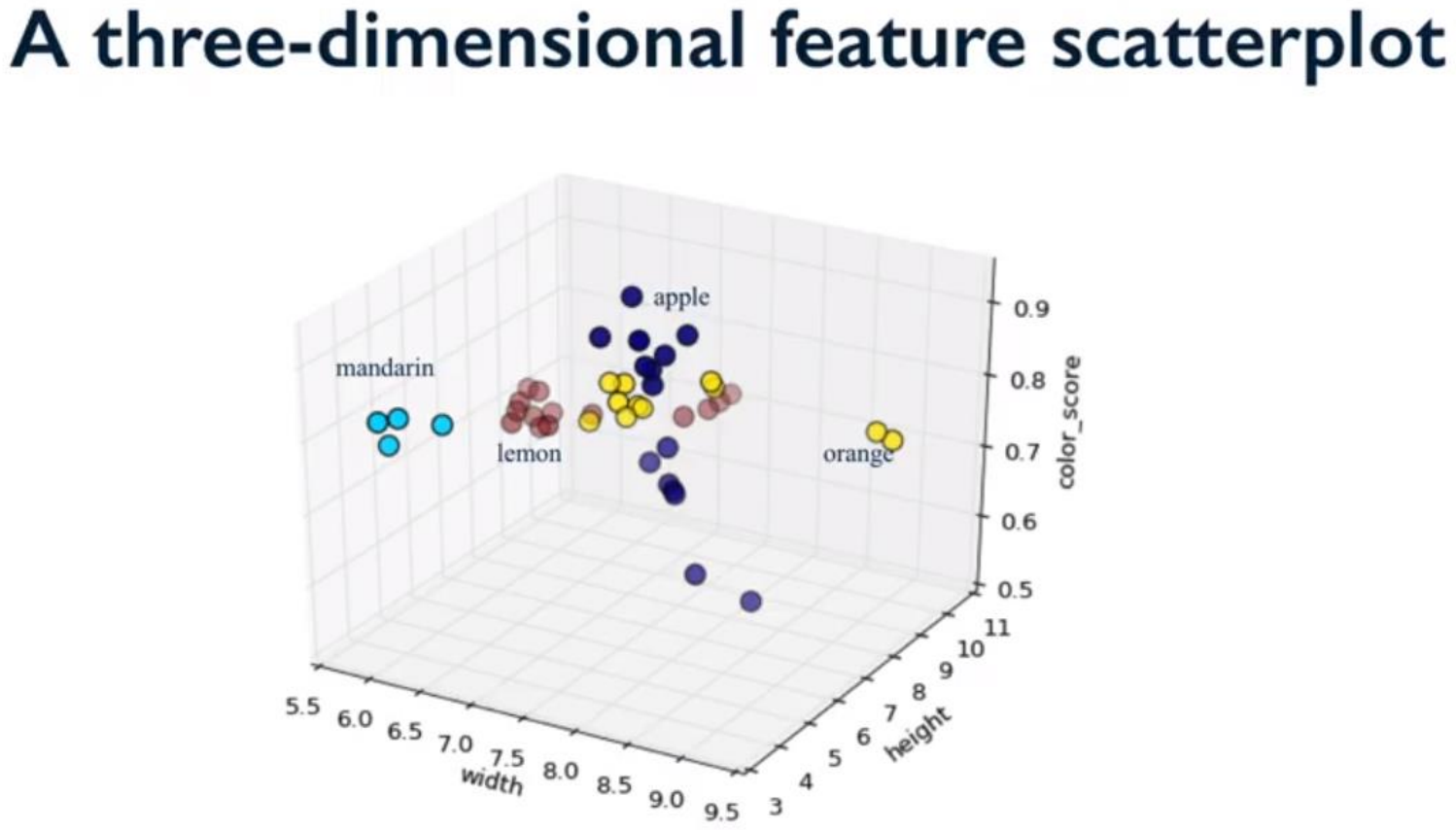
Always Remember to inspect your Data

Examples of incorrect or missing feature values

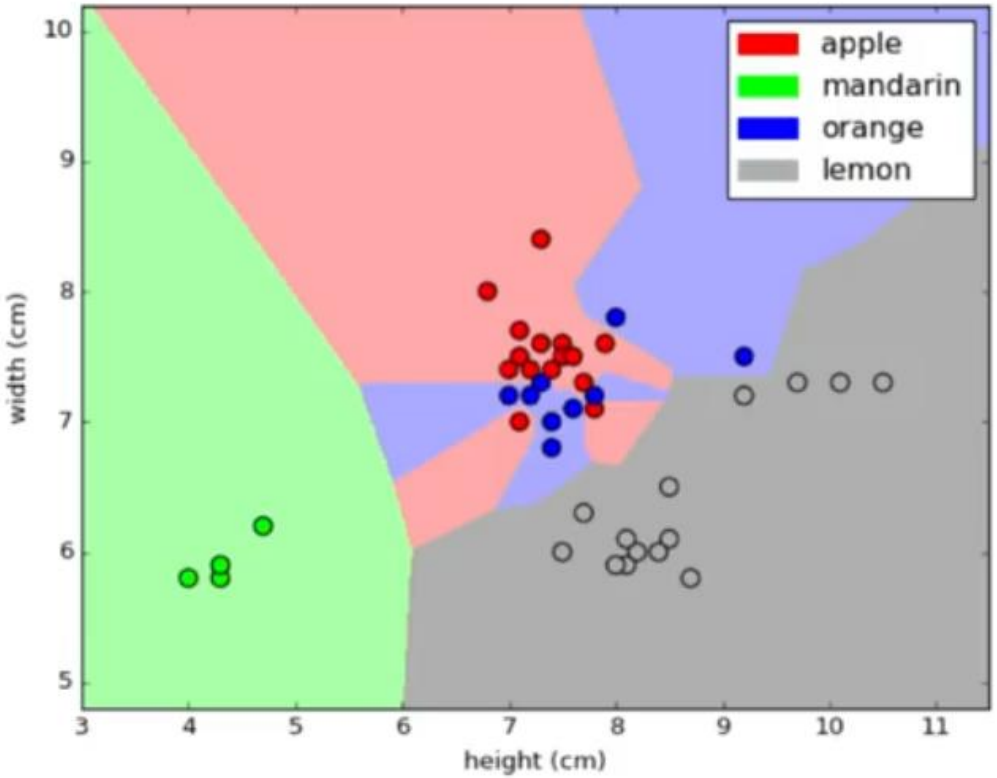
	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59
2	1	apple	granny_smith	176	7.4	7.2	192
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79
5	2	mandarin	apple	80	5.8	4.3	0.77
6	2	mandarin	mandarin	80	5.9	4.3	0.81
7	2	mandarin	mandarin	76	5.8	4.0	0.81
8	1	apple	braeburn	178	7.1	7.8	0.92
9	1	apple	braeburn		7.4	7.0	0.89
10	1	apple	braeburn		6.9	7.3	0.93
11	1	apple	braeburn		7.1	7.6	0.92
12	1	apple	braeburn		7.0	7.1	0.88
13	1	apple	golden_delicious		7.3	7.7	0.70
14	1	apple	golden_delicious	152	7.6	7.3	0.69

Example taken from Applied Machine Learning in Python course by the University of Michigan
https://www.coursera.org/learn/python-machine-learning

Plot your Data



Example taken from Applied Machine Learning in Python course by the University of Michigan
https://www.coursera.org/learn/python-machine-learning



Fruit dataset
Decision boundaries
with k = 1

Example taken from Applied Machine Learning in Python course by the University of Michigan
https://www.coursera.org/learn/python-machine-learning

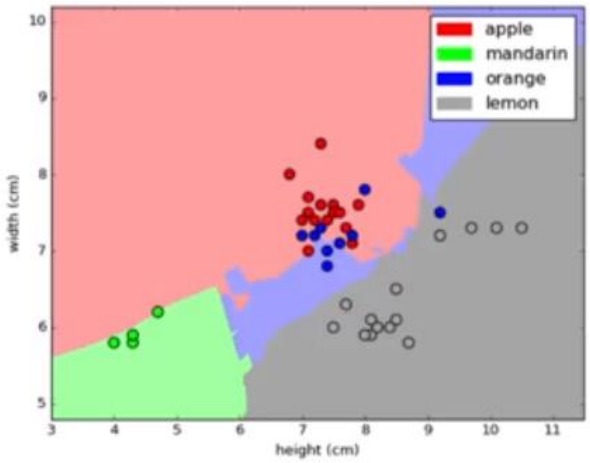
Generalization, Overfitting and Underfitting

- Generalization ability refers to an algorithm's ability to give accurate predictions for new, previously unseen data.
- Assumptions:
  - *Future unseen data (test set) will have the same properties as the current training sets.*
  - *Thus, models that are accurate on the training set are expected to be accurate on the test set.*
  - *But that may not happen if the trained model is tuned too specifically to the training set.*
- Models that are too complex for the amount of training data available are said to overfit and are not likely to generalize well to new examples.
- Models that are too simple, that don't even do well on the training data, are said to underfit and also not likely to generalize well.

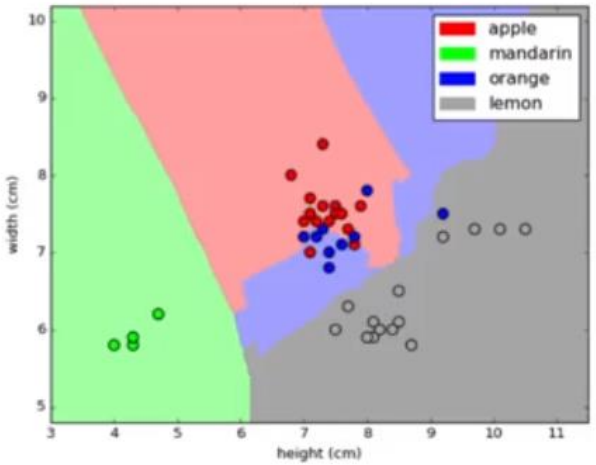
Generalization, Overfitting and Underfitting

- Generalization ability refers to an algorithm's ability to give accurate predictions for new, previously unseen data.
- Assumptions:
  - Future unseen data (test set) will have the same properties as the current training set
  - This, models that are accurate on the training set are expected to be accurate on the test set
  - But that may not happen if the trained model is tuned too specifically to the training set.
- Models that are too complex for the amount of training data available are said to **overfit** and are not likely to generalize well to new data instances.
- Models that are too simple, that do not even do well on the training data, are said to **underfit** and also not likely to generalize well.

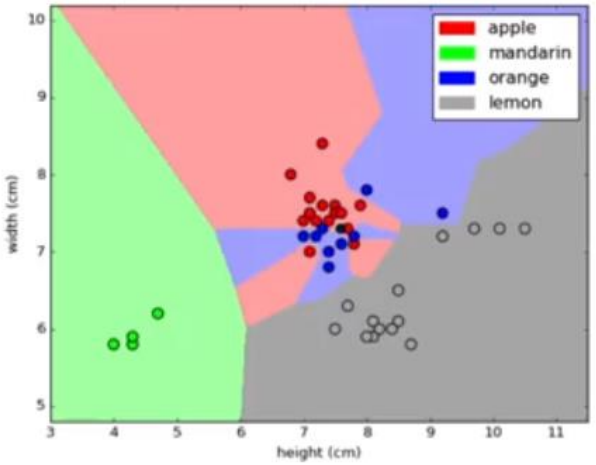
Overfitting with k-NN classifiers



K=10

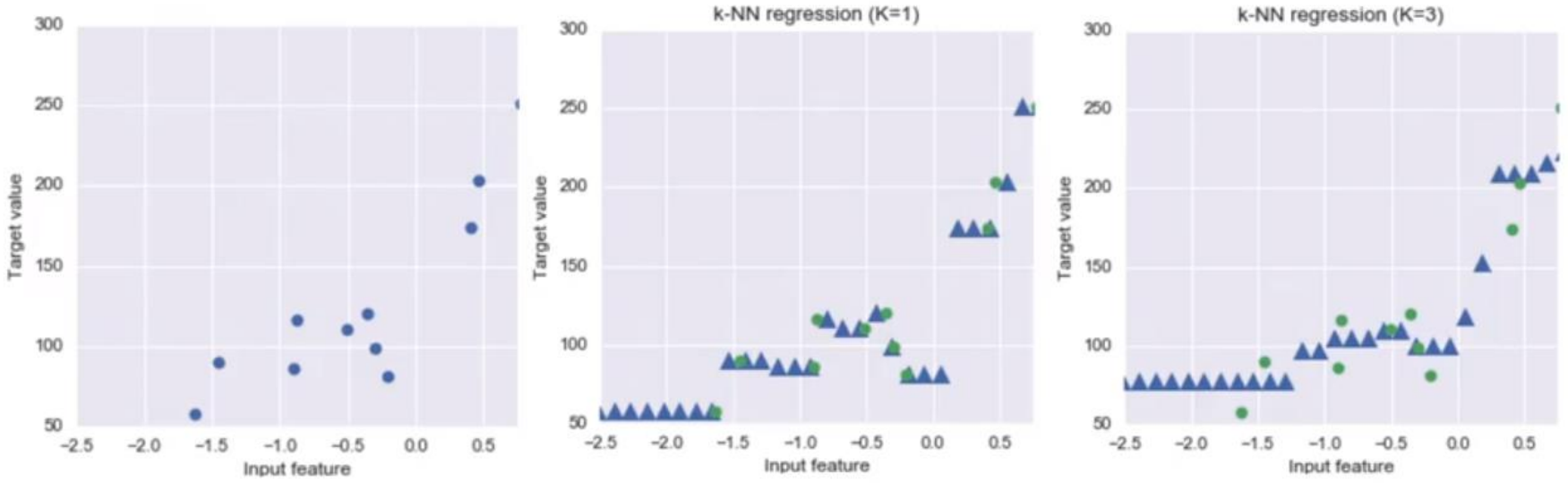


K=5



K=1

k-Nearest neighbors regression



PROJECT

What is the Project About?

In this assignment, you are asked to **develop** a Data Analytics Service. Besides a few requirements that you must meet, the specification of the application you'd build is **deliberately left open**. You are expected to **develop a plan** and **execute it** with your **group members** all throughout this assignment period.

Mentoring Meetings

You can arrange meetings with the tutors from Week 10. Before the final demo, you are required to have two meetings. Each meeting will be assessed. During the meetings, your mentor (tutor) will give feedback on your work in progress. So utilise the time as much as you can.

- Meeting One - more or so complete design documentation (10% of the total mark).
- Meeting Two - an early implementation of the Service, demo of work in progress (10% of the total mark).

What you need to do

- Come up with a scenario
- Finding data-sources and Data Integration (if you are using more than two data sources)
- Building an analytical module (select a Machine learning model and train it using your existing data set) to fulfil the scenario
- Designing and Build a REST API for your service (you need to also consider authentication)
- Designing a Simple Client with GUI (feel free to use what ever you are familiar with whether it is a simple HTML-javascript, ASP, php, JSP, JSF, or even window-based interfaces)

Service Examples

- Property Price Prediction
- White Hackers Expertise prediction
- Movies Box Office revenue prediction

How Will We be Marked for this Project

- Mentoring Meeting One - 10%
- Mentoring Meeting Two - 10%
- Week 13 - 70% (demo), 10% (group work)

Useful Resources

- Mathematician hacking dating site https://www.wired.com/2014/01/how-to-hack-okcupid/
- https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7
- https://towardsdatascience.com/building-improving-a-k-nearest-neighbors-algorithm-in-python-3b6b5320d2f8
- https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/
- https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/